

IOWA STATE UNIVERSITY

Digital Repository

Mechanical Engineering Conference Presentations,
Papers, and Proceedings

Mechanical Engineering

8-2013

Virtual Hand Representations to Support Natural Interaction in Immersive Environment

Meisha N. Rosenberg
Iowa State University, meishar@iastate.edu

Judy M. Vance
Iowa State University, jmvance@iastate.edu

Follow this and additional works at: http://lib.dr.iastate.edu/me_conf

 Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Rosenberg, Meisha N. and Vance, Judy M., "Virtual Hand Representations to Support Natural Interaction in Immersive Environment" (2013). *Mechanical Engineering Conference Presentations, Papers, and Proceedings*. Paper 18.
http://lib.dr.iastate.edu/me_conf/18

This Conference Proceeding is brought to you for free and open access by the Mechanical Engineering at Digital Repository @ Iowa State University. It has been accepted for inclusion in Mechanical Engineering Conference Presentations, Papers, and Proceedings by an authorized administrator of Digital Repository @ Iowa State University. For more information, please contact digirep@iastate.edu.

DETC2013-13166

VIRTUAL HAND REPRESENTATIONS TO SUPPORT NATURAL INTERACTION IN IMMERSIVE ENVIRONMENTS

Meisha Rosenberg

Department of Mechanical Engineering
Virtual Reality Applications Center
Iowa State University
Ames, Iowa, USA
meishar@iastate.edu

Judy M. Vance

Department of Mechanical Engineering
Virtual Reality Application Center
Iowa State University
Ames, IA 50011 U.S.A.
jmvance@iastate.edu
ASME Fellow

ABSTRACT

Immersive Computing Technology (ICT) offers designers the unique ability to evaluate human interaction with product design concepts through the use of stereo viewing and 3D position tracking. These technologies provide designers with opportunities to create virtual simulations for numerous different applications. In order to support the immersive experience of a virtual simulation, it is necessary to employ interaction techniques that are appropriately mapped to specific tasks. Numerous methods for interacting in various virtual applications have been developed which use wands, game controllers, and haptic devices. However, if the intent of the simulation is to gather information on how a person would interact in an environment, more natural interaction paradigms are needed. The use of 3D hand models coupled with position-tracked gloves provide for intuitive interactions in virtual environments. This paper presents several methods of representing a virtual hand model in the virtual environment to support natural interaction.

1. INTRODUCTION

Immersive Computing Technology (ICT) provides users in a variety of domains the ability to create high fidelity simulations of real-world situations that are either too dangerous or too costly to develop. For instance, medical professionals can practice complex surgical procedures without risking the lives of their patients. Additionally, pilots can perform risky maneuvers before stepping foot in a plane. Designers can evaluate and optimize design concepts before assembly lines and working prototypes are created at high cost. There are many tools, devices, and techniques that support ICT experiences.

Interactive ICT simulations can be created with small, low-cost desktop environments, Head-Mounted Displays (HMDs), and immersive projection environments. Each of these environments utilizes a variety of input devices to facilitate user interaction within the immersive experience.

Wands, game controllers, and haptic devices provide the user with the ability to move and interact in virtual environments. Because these devices present abstract ways to interact with the environment they inherently insert an additional cognitive load on the user. Additionally, there is an initial learning curve that also needs to be taken into account. For example, to navigate through the environment, a person might push a button, move a wand, roll a track ball or one of many other possible actions. This additional load on the cognitive resources can potentially have a negative impact on the user's response to the virtual environment and ability to interact within the virtual environment. Our goal is to make interactions within the virtual environment as seamless and natural as possible in order to focus the user's attention on the task at hand and not on the interaction methods.

Studies have shown that hands play a key role in how we are able to communicate and interact in virtual

environments [7, 14]. It is for this reason that virtual representations of hands coupled with physical data gloves show great potential as a tool for manipulation in a virtual environment.

The focus of this paper is on examining various representations of virtual hands in virtual environments to support natural interaction. Section 2 will discuss fundamentals of input devices for virtual environments. Section 3 will discuss various methods of virtual hand models. Section 4 presents a summary and section 6 suggests further opportunities for research.

2. CURRENT INPUT DEVICES FOR INTERACTION

2.1 Mouse and Keyboard

Since the early days, the mouse and keyboard have been the typical input devices used to interact with desktop computers. In fact, these devices are so commonly used that, until the recent availability of touch screens, many people considered these devices to be a required part of every computer. With the advent of stereo-capable monitors, users can design 3D environments to experience desktop or “fishtank” VR. Research by Karat et al. [28] shows that the use of a mouse in desktop configurations yields a lower performance and applies a higher cognitive load than even touch screen interactions. Its two-dimensional boundaries create many limits when interacting in a three-dimensional environment and therefore it is difficult to take full advantage of the abilities of the software or the advantages of the simulation.

2.2 Game Controllers and Wands

The introduction of a 3D position-tracking device allows for tracking of a multitude of potential interaction devices in 3D space. Popular methods of interacting in more immersive virtual environments, such as projection screen or head mounted display environments (HMD), include the use of game controllers, wands (Fig. 1), haptic devices or gloves (Fig. 2). With the addition of a 3D position tracking system, some of these interaction devices can be used with desktop VR systems as shown in Figure 3. Each of these interaction devices offers different advantages.

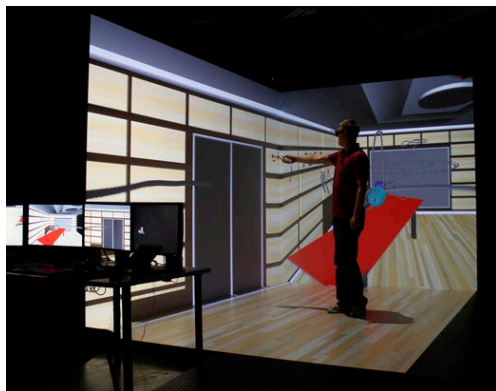


Figure 1. IMMERSIVE PROJECTION ENVIRONMENT WITH WAND



Figure 2. HEAD MOUNTED DISPLAY WITH GLOVE

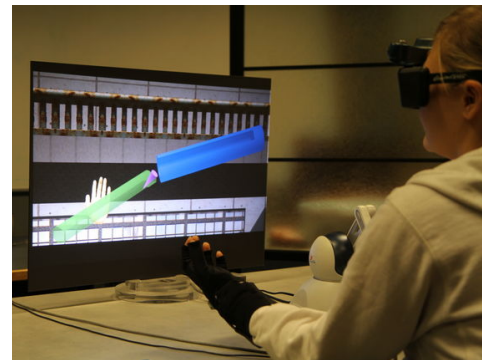


Figure 3. DESKTOP COMPUTER ENVIRONMENT WITH REAR-PROJECTED SCREEN, GLOVE, AND HAPTIC DEVICE

The game controller is as familiar to the millennial generation as the mouse is to generation X. Coupled with a position tracker, it can easily be implemented to interact with a 3D immersive environment. Buttons and scroll wheels can be programmed to execute actions in the simulation. However, despite the familiarity among current users, it is not a device that is easily used by older generations or those unfamiliar to console gaming. Wands are similar to game controllers. They contain multiple buttons and other sensors that can be integrated as input to the immersive environment.

With respect to both game controllers and wands, mappings of the controller sensors can vary widely from one controller to the next. The generality of the controller allows the environment designer to adapt the various buttons and joysticks to their specific tasks, however; this flexibility can

result in widely varying task mappings from one application to the next. One button that supports navigation in one application could be the EXIT button in another application. Additionally many of the buttons are similar and therefore the user must remember the location of each button and its unique function.

2.3 Haptic Devices

Haptic devices offer the unique ability for a user to receive force feedback information about the objects that they are interacting with in the virtual environment. Haptic devices can take many forms depending on the environment they are being used in. Smaller pen-like devices are commonly used in desktop environments (Fig. 4), allowing the user to interact with fine details of the environment. Larger devices can be used in bigger environments such as immersive projection systems. The user generally holds some sort of handle as an interface to the haptic device, much like a wand interface.



Figure 4. HAPTIC DEVICE

2.4 Data Gloves

Data gloves can also be used to interact with a 3D simulation (Fig. 5). Sensors in the glove report the position of each finger as the user interacts with the simulation. When coupled with a position tracking system, a computer generated model of the hand can represent the user's hand interacting in the environment.



Figure 5. 5DT DATA GLOVE WITH 3D POSITION TRACKER

The first data gloves were created in the 70's [1] for the purpose of tracking body parts for computer animations. Although the technology has developed significantly over the decades, the basic architecture of the glove is essentially the same. Data gloves are typically made of a tight fitting Lycra

material with sensors sewn into them along the length of each of the fingers. They can include anywhere from 1 to 30 different sensors. An increased number of sensors on the gloves means that more detailed positioning data can be acquired. Figure 5 shows an example of a 5DT data glove with 5 sensors running along the length of the fingers.

To be an effective input device, the hand must be tracked in 3D space and the bend angles of the fingers must be reported. A 3D position tracking system is generally attached to the hand providing data to the simulation on the location and orientation of the hand in the simulated 3D space. Additional sensors on the glove provide information on the bending of each finger. Once the bend sensor data has been collected, it is often filtered using a Kalman Filter [6] in order to minimize the noise produced from the signal. The signal is then mapped to the 3D hand model in the virtual environment. The method for mapping this data to the hand model can vary based on the method that is used for rendering the hand model.

Recently, depth cameras have been developed which can be used to track a person's position in space. These near-infrared cameras rely on the transmission and reflection of light to determine where an object is in space. This method can be used to track fingers but is not well developed or robust at this time. In the mid 90's Heap et al. [15] created a system that utilized Point Distribution Models to track an unmarked hand in real-time. This system however had trouble handling occluded parts of the user's hand, as well as ambiguities in planar rotation. In 2008, de La Gorce et al. [10] proposed a system that removes many of these issues as well as provides textures and shading of the hand model. His system minimizes an objective function in order to estimate the texture of the hand and the shadowing in real-time. Hackenberg et al. [9] refined the technique further by implementing a time-of-flight camera to capture infrared signals, eliminating the challenges caused by lighting. Although these systems have evolved dramatically over the past two decades, they are still not capable of capturing all user data seamlessly.

In order to interact using data gloves, a virtual hand model must be present in the environment. In desktop, immersive projection and head mounted displays, it is key that the user sees a computer-generated image of his/her hand in the immersive environment. In the desktop configuration, the user is looking into the 3D environment and therefore needs to see the hand moving in the space behind the physical screen. In the immersive projection screen environment, the physical hand can occlude images projected on the screens. The result is that the hand can never go under or in a computer generated object. Therefore, a virtual hand model is required in this configuration. When using a head mounted display, no real parts of the user's body are shown in the display so a virtual hand model is required. The rest of this paper will describe various ways to implement a virtual hand model.

3. METHODS FOR RENDERING 3D HAND MODELS

There are many different methods that have been developed to create a hand model for display in a virtual environment. These various methods all have their unique benefits and drawbacks. It is up to the designers of the simulation to decide which method is most effective for their desired interaction.

3.1 Fixed Configuration Models

There are several situations where it is not necessary to have a dynamic virtual model of the hand. In situations where the focus is not on the actual hand configuration, a fixed geometric model of a hand can be used. These models can be created using advanced modeling software. In some situations, the purpose of the simulation is not necessarily for the user to use their hands to interact with the environment, but rather to explore their environment in other ways. The presence of a virtual hand model in the scene provides the user with the ability to more effectively gauge distances and therefore better perceive the details of their environment.

There are two ways to implement a fixed configuration model. First and most simply, a single geometric model of the hand can be created. This single model is used throughout the simulation with no change. It can be thought of as a 3D cursor in the shape of a hand (Fig. 6). This method was implemented by Seth et al. [16] in the SHARP software. Another instance where this method would be appropriate is discussed by Ulrich [5]. The application consists of training medical technicians in the insertion of a needle into a patient. In this specific simulation, the focus of the simulation is on the orientation of the needle and how it interacts with the patient, not necessarily on the configuration of the hand and fingers of the trainee.

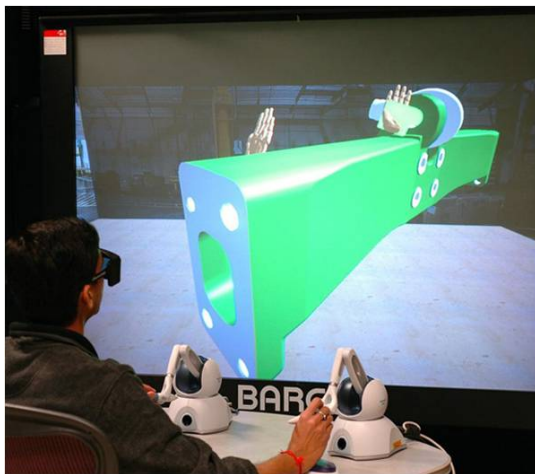


Figure 6. FIXED CONFIGURATION OF HAND MODEL [18]

Another way to implement fixed configuration models is to have a series of different configurations represented by geometric models. When the hand sensor indicates a change in hand configuration, the appropriate geometric hand model is loaded into the scene. One example of this implementation is shown in the Virtual Factory [17] (Fig. 7).

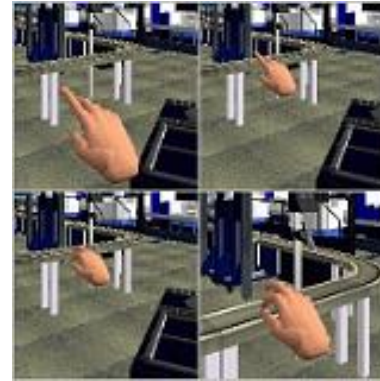


Figure 7. FIXED CONFIGURATION OF HAND MODELS [19]

Another example of where a series of fixed hand models would be appropriate is in an immersive environment for training people in the use of sign language [2, 4]. In basic sign language, there are a limited number of hand positions that allow a user to communicate. A comprehensive fixed set of hand models in given positions can be used to effectively communicate the intent of the simulation.

3.2 Real-Time Deformable Mesh Models

Real-time deformable mesh models are one of the most realistic methods currently being implemented. The level of realism is so great that in some cases hand models are created using a Magnetic Resonance Image (MRI) of a real human hand. Software is then used to map a mesh to the MRI creating a virtual representation of the physical hand [11, 22]. Deformable mesh models are used in a large variety of ICT simulations. While many of these applications are used for general surgery simulations, they are also quite appropriate in the simulations of deformable mesh hands. Because it is not necessary to model the physics of tissue in a hand model, applying these methods to hand becomes even more simplified. These deformable mesh models can be made so they can be updated in real-time [25] or pre-computed [27]. Methods of supporting the deformation include spring-mass models, linked volumes, and finite element methods.

3.2.1 Spring-mass Models

Generally, Spring-mass models consist of a series of discrete point masses connected by springs or dampers in a mesh [19]. The following second order differential equation defines the force applied to each node by the nodes that are attached to it:

$$F = M\ddot{x} + C\dot{x} + Kx$$

Where M represents the mass matrix of the object, C represents the damping coefficient matrix, K represents the stiffness coefficient matrix and F represents the forces acting on the object. When one node moves, for instance a finger is bent, the other forces on the other nodes are calculated such that the system remains in equilibrium [11]. This method of

computation allows spring-mass models to provide a high quality of rendering while maintaining a low computation cost. Spring-mass models have been used consistently since the early 90's and are still extremely relevant in rendering deformable hand models. They are commonly used in applications that require the use of a hand model that moves dynamically in real-time. Such applications include surgery simulations [20] as well as product assembly tasks. In these simulations the positioning of the hands plays a critical role in how the simulation is completed. Spring-mass models require an extensive understanding of the underlying mathematical principles and require more rendering capabilities than are necessary in all situations.

3.2.2 Linked Volumes Model

Linked Volumes are similar to Spring-mass models in that they are also comprised of a series of masses linked together. Unlike the Spring-mass hand models however, the Linked Volumes hand models links together each of the masses without the existence of springs. There are various methods in which these nodes can be linked together. One method employs linking each mass to its six neighbors, creating a cubic structure [23]. Although this method requires more computational power, it allows the user to control a hand model in a more interactive manner without the hours of preprocessing that was previously required. Other methods create linkages across only the surface of the model, creating a chain mail surface where each linkage is connected to eight others [18,21]. This is less computationally demanding because only the surface of the model is rendered. This makes this method ideally suited for deformable mesh hand models however, not useful for any applications which require an entire volume to be rendered.

3.2.3 Finite Element Methods

Finite Element Methods (FEMs) also create a mesh of nodes throughout the volume of a model over which a series of differential equations related to the displacement of each of the nodes are then solved. The nodes are arranged in a tetrahedral manner. Material properties are provided to produce a realistic simulation of deformation [13]. In these implementations, a voxel-based, FE mesh is drawn to represent the desired model. The linear elastic energy of the model is computed based on the displacement of each of the vertices [26].

Finite Element Methods are quite successful in modeling large volumetric geometries, and computing the effects that large deformations have on the model geometry. However because of the number of nodes used to create these models, they are extremely computationally expensive. This means that the deformations must be pre-computed, or a crude hand model with few nodes must be used. Additionally, the system used to compute the deformation of the nodes using FEM must have high processor speeds, meaning that they are unable to be applied in all environments.

3.3 Kinematic Models

In order to minimize the impact on rendering a high definition virtual model some applications create a hand by linking a

series of geometric shapes in a kinematic chain. This method allows the hand model to be rendered in real time without significant rendering requirements. Additionally, mapping the data output from the sensors of the glove to the various features of the hand model becomes a trivial task. The joint angle of each joint is passed to the application and used to determine the new position of each rigid link. The links are then connected together to form a complete representation of the hand, similar to the manner in which the various joints of a robot work together. The more sensor data that is available from the glove, the more individual components of the hand can be mapped. However, because of the simplicity of the shapes used to create the model of a hand, it is necessary to apply a texture map to the model in order to attain a higher level of realism. This technique was implemented quite successfully by Wan et al. in 2004 [29]. The use of the texture map resulted in a relatively realistic and immersive hand model with limited processing demands.. This method does present some challenges however, when used to render objects that are of a more complex geometric representation.

4 SUMMARY

Our hands play a key role in our interaction with our physical environment. Many current forms of immersive computing technologies do not take full advantage of our interactions in physical space. Instead they employ tools and devices such as mice and keyboards, wands, game controllers and haptic devices as methods for interaction in virtual environments. The unnatural nature of these devices imposes an added cognitive load on the user and limits their abilities to interact naturally in simulations.

Data gloves are an alternative method for interacting within virtual environments that are much more natural and in turn create a greater sense of immersion for users. By interacting with a virtual environment with a data glove, users avoid the additional cognitive load present when using other interaction devices.

A realistic deformable hand model is a key part of the immersive experience. There are many different methods that can be employed to create such a hand model. Each method comes with its own benefits as well as drawbacks and is therefore suited to different applications. Fixed models can help to create a sense of immersion in some situations, however they cannot be updated in real-time. Therefore they are not ideally suited for situations in which the motion of the hand model needs to be directly mapped to the motion of the user's hand. Deformable mesh models typically provide quite realistic renderings of hand models that can be mapped to the physical motions of the user. This creates a much more immersive experience, however it also requires increased rendering capabilities and processing speed and is therefore not well suited for all applications. Kinematic models allow the user's physical motions to be updated in real time in the virtual simulation, however; they are not as realistic as many of the models created through deformable meshes.

TABLE 6. SUMMARY OF RENDERING METHODS

Modeling Method	Rendering Time	Realism
Fixed Configuration Models	Pre-computed	Low
Spring-Mass Models	Real-time	High
Linked Volumes Models	Real-time	High
Finite Element Models	Pre-computed	High
Kinematic Models	Real-time	Low to medium

5 CONCLUSIONS

As new processing capabilities emerge, it becomes possible to create virtual environments that utilize more realistic hand models, improving the immersive experiences of the users and allowing for more natural interactions to take place. In the future, as improvements in near-infrared technology occur, the underlying data that drives the hand model will change, but the basic need for virtual hands in the environment will still be present. Understanding the various methods of virtual hand model simulation and the relationship to the objectives of the simulation will continue to be a key contributor to effective design of virtual environments.

Acknowledgement

This material is based upon work supported by the National Science Foundation under grant #CMMI-0928774. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

- [1] Dipietro, L., Sabatini, A. M., & Dario, P. (2008). A survey of glove-based systems and their applications. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(4), 461-482.
- [2] Crasborn, O., Sloetjes, H., Auer, E., & Wittenburg, P. (2006, May). Combining video and numeric data in the analysis of sign languages within the ELAN annotation software. In *Proc. LREC 2006 workshop on representation & processing of sign languages* (pp. 82-87).
- [3] Terzopoulos, D., Platt, J., Barr, A., & Fleischer, K. (1987, August). Elastically deformable models. In *ACM Siggraph Computer Graphics* (Vol. 21, No. 4, pp. 205-214). ACM.
- [4] Fillbrandt, H., Akyol, S., & Kraiss, K. F. (2003, October). Extraction of 3D hand shape and posture from image sequences for sign language recognition. In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures* (p. 181). IEEE Computer Society.
- [5] Ullrich, S., & Kuhlen, T. (2012). Haptic Palpation for Medical Simulation in Virtual Environments. *Visualization and Computer Graphics, IEEE Transactions on*, 18(4), 617-625.
- [6] Welch, G. F. (2009). History: The use of the kalman filter for human motion tracking in virtual reality. *Presence: Teleoperators and Virtual Environments*, 18(1), 72-91.
- [7] Cornelius, C. J., & Hayes, C. C. (2011, October). How important are hand images for conveying gestures in virtual design tasks?. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on* (pp. 2045-2050). IEEE.
- [8] Cover, S. A., Ezquerro, N. F., O'Brien, J. F., Rowe, R., Gadacz, T., & Palm, E. (1993). Interactively deformable models for surgery simulation. *Computer Graphics and Applications, IEEE*, 13(6), 68-75.
- [9] Hackenberg, G., McCall, R., & Broll, W. (2011, March). Lightweight palm and finger tracking for real-time 3D gesture control. In *Virtual Reality Conference (VR), 2011 IEEE* (pp. 19-26). IEEE.
- [10] de La Gorce, M., Paragios, N., & Fleet, D. J. (2008, June). Model-based hand tracking with texture, shading and self-occlusions. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (pp. 1-8). IEEE.
- [11] Kurihara, T., & Miyata, N. (2004, August). Modeling deformable human hands from medical images. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (pp. 355-363). Eurographics Association.
- [12] Meier, U., Lopez, O., Monserrat, C., Juan, M. C., & Alcaniz, M. (2005). Real-time deformable models for surgery simulation: a survey. *Computer methods and programs in biomedicine*, 77(3), 183-197.
- [13] Bro-Nielsen, M. (1996). Surgery simulation using fast finite elements. In *Visualization in Biomedical Computing* (pp. 529-534). Springer Berlin/Heidelberg.
- [14] Mohler, B. J., Creem-Regehr, S. H., Thompson, W. B., & Bühlhoff, H. H. (2010). The effect of viewing a self-avatar on distance judgments in an HMD-based virtual environment. *Presence: Teleoperators and Virtual Environments*, 19(3), 230-242.
- [15] Heap, T., & Hogg, D. (1996, October). Towards 3D hand tracking using a deformable model. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on* (pp. 140-145). IEEE.
- [16] Seth, A., Su, H.-J., Vance, J. M., "SHARP: A system for haptic assembly and realistic prototyping", ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Sept 10-13, 2006, Philadelphia, PA, DETC2006-99476.

- [17] Kelsick, J., Vance, J. M., Buhr, L., and Moller, C., "Discrete event simulation implemented in a virtual environment", *ASME Journal of Mechanical Design*, 125(3), 2003, pp. 428 - 433.
- [18] Gibson, S. F. (1997, April). 3D chainmail: a fast algorithm for deforming volumetric objects. In *Proceedings of the 1997 symposium on Interactive 3D graphics* (pp. 149-ff). ACM.
- [19] Deussen, O., Kobbelt, L., & Tücke, P. (1995). *Using simulated annealing to obtain good nodal approximations of deformable bodies*. Bibliothek der Universität Konstanz.
- [20] Kuhn, C., Kühnapfel, U., Krumm, H. G., & Neisius, B. (1996). Karlsruhe Endoscopic Surgery Trainer-A Virtual Reality-based Training System for Minimally Invasive Surgery. In *MMVR '97*. IOS Press Ohmsha, Washington DC. Mazura 1997b Mazura, A.
- [21] Gibson, S., Samosky, J., Mor, A., Fyock, C., Grimson, E., Kanade, T., ... & Sawada, A. (1997). Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback. In *CVRMED-MRCAS'97* (pp. 367-378). Springer Berlin/Heidelberg.
- [22] Heap, T., & Hogg, D. (1996, March). 3D deformable hand models. In *Gesture Workshop, York, UK* (pp. 131-139).
- [23] Frisken-Gibson, S. F. (1999). Using linked volumes to model object collisions, deformation, cutting, carving, and joining. *Visualization and Computer Graphics, IEEE Transactions on*, 5(4), 333-348.
- [24] Brown, J., Sorkin, S., Latombe, J. C., Montgomery, K., & Stephanides, M. (2002). Algorithmic tools for real-time microsurgery simulation. *Medical Image Analysis*, 6(3), 289-300.
- [25] Picinbono, G., Delingette, H., & Ayache, N. (2000). Real-time large displacement elasticity for surgery simulation: Non-linear tensor-mass model. In *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2000*. Springer Berlin/Heidelberg.
- [26] Kieve, E., Girod, S., & Girod, B. (1996). Craniofacial surgery simulation. In *Visualization in Biomedical Computing* (pp. 541-546). Springer Berlin/Heidelberg.
- [27] Koch, R. M., Gross, M. H., Carls, F. R., von Büren, D. F., Fankhauser, G., & Parish, Y. I. (1996, August). Simulating facial surgery using finite element models. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (pp. 421-428). ACM.
- [28] Karat, J., McDonald, J. E., & Anderson, M. (1986). A comparison of menu selection techniques: touch panel, mouse and keyboard. *International Journal of Man-Machine Studies*, 25(1), 73-88.
- [29] Wan, H., Luo, Y., Gao, S., & Peng, Q. (2004, June). Realistic virtual hand modeling with applications for virtual grasping. In *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry* (pp. 81-87). ACM.